

# Simulación de Sistemas: FHP-I Lattice-Gas

Julian Nicastro, Sespede Braulio

11 de abril de 2017

## Resumen

Resolver y predecir el flujo de fluidos es una tarea importante para ingenieros, físicos y matemáticos. Puede ser una tarea matemáticamente compleja para dominios con espacios de geometría compleja. Por este motivo, es deseable encontrar modelos que representen dicho comportamiento en vez de resolverlo con las ecuaciones originales de Navier-Stokes. Con este propósito se diseñaron modelos como el del **lattice gas** y lattice Boltzmann. En este trabajo se estudia la implementación del modelo **FHP-I**, de la familia de lattice gas.

## 1. Introducción

Un autómata celular de lattice gas, se caracteriza, como el resto de los autómatas celulares, por estar compuesto por partículas dispuestas en un espacio discreto. Dichas partículas se caracterizan por poseer distintos estados, en el caso del lattice gas, tienen distintas velocidades de igual magnitud pero distinta dirección.

La evolución en el tiempo del sistema es de pasos discretos, donde en cada paso se puede determinar el estado en una posición por el estado de la posición misma y la de sus vecinos en dicho paso.

En cada paso, el estado de una posición esta determinado por la existencia de partículas sobre la misma. A su vez, en cada paso, se llevan a cabo dos procesos: la propagación y colisión.

En el paso de propagación, cada partícula se moverá a la posición vecina determinada por el vector velocidad de la partícula y posición actual.

En el paso de colisión, se utilizan reglas sencillas de colisión para determinar que sucede si múltiples partículas han de trasladarse a la misma posición a la vez o si una partícula ha de chocar con un obstáculo.

En el primer modelo, por Pomeau y de Pazzis, llamado HPP, las partículas solo pueden trasladarse horizontal y verticalmente, y en caso de colisionar pasan a moverse perpendicularmente a su dirección original. La consecuencia de estas restricciones hacen que los vórtices producidos por dicho modelo tengan forma cuadrada.

Posteriormente, se presentó otro modelo por Uriel Frisch, Brosl Hasslacher e Yves Pomeau, donde se permite el movimiento diagonal a través de una grilla hexagonal. Dicho modelo se conoce como FHP en honor a sus investigadores. Según la variante existen 6 o 7 estados posibles por posición. En cualquiera de las variantes, 6 de los estados representan la velocidad con la cual se desplazan las partículas (misma magnitud pero distintas direcciones). En algunas variantes (llamadas FHP-II y FHP-III), se toma un estado adicional para indicar el reposo de una partícula. Dichas partículas no se propagan a sus vecinos pero pueden colisionar con otras partículas en cada paso de darse el caso.

Si bien el modelo es sencillo basta para simular el comportamiento complejo de fluidos.

## 2. Modelo

La idea de los algoritmos de la familia de lattice gas es que si las propiedades esenciales de los fluidos reales se mantienen a nivel microscópico, entonces el comportamiento físico a nivel microscopio será correcto. El modelo toma dos asunciones:

- Conservación de las partículas en cada paso.
- Conservación del momento en cada paso.

En particular, el modelo FHP-I se caracteriza por:

- Grilla hexagonal de nodos.
- Cada nodo tiene 6 celdas representando partículas con 6 velocidades distintas.
- Las celdas pueden ser ocupadas por a lo sumo una partícula.
- Todas las partículas poseen la misma masa.
- Solo las partículas dentro un nodo son involucradas en la colisión.

El modelo básico de FHP contempla colisiones entre dos y tres partículas. En cualquier otro caso continúan su trayectoria ya que se conserva el momento según el modelo planteado.

Si dos partículas trasladándose en direcciones opuestas se encuentran en el mismo nodo, el par de partículas rota su vector velocidad 60 grados en sentido de las manecillas del reloj, o en el sentido contrario, de manera pseudoaleatoria. En caso de que la colisión sea entre tres partículas, colisionan de manera que se “invierta” la configuración de las posiciones entre partículas. Ver figura 1.

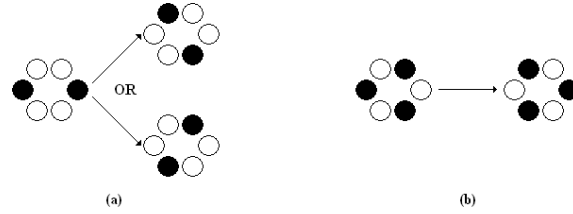


Figura 1: Posibles colisiones entre partículas en FHP-I. Los puntos negros representan la dirección de los vectores velocidades de las partículas que colisionan y los puntos blancos aquellas direcciones que no han sido ocupadas. (a) Colisión entre dos partículas. (b) Colisión entre tres partículas.

### 3. Implementación

Para representar los nodos se utilizó un arreglo tridimensional en donde las primeras dos coordenadas representan la posición del nodo y la última coordenada es un vector con las 6 posibles direcciones, en donde se indica con un 0 si no fue utilizada y con un 1 en caso contrario.

Se tomo la siguiente convención para representar la posición de la dirección en el vector que indica el uso o no de las direcciones:

- 0 = DERECHA
- 1 = ARRIBA-DERECHA
- 2 = ARRIBA-IZQUIERDA
- 3 = IZQUIERDA
- 4 = ABAJO-IZQUIERDA
- 5 = ABAJO-DERECHA

Por otro lado se representaron las paredes y obstáculos con un arreglo bidimensional en donde se indica con un 1 si dicha posición posee un obstáculo y con un 0 en caso contrario.

La simulación implementada usando el modelo FHP procede en tres pasos.

El primer paso de la simulación es la inicialización con el tamaño del dominio, la cantidad de pasos, el tamaño/posición de la barrera/paredes y el tamaño del subdominio para el cálculo de velocidades promedio (en el output no se generan los vectores velocidad para todos los nodos). Para la generación de partículas se implementó la primer opción propuesta por la cátedra: cada 4 pasos de simulación se inyectan 3 partículas en cada sitio del borde izquierdo con velocidades en 3 direcciones distintas.

Como se comentó anteriormente la simulación tiene en cuenta obstáculos (barrera de tamaño  $L$  y paredes en la parte superior e inferior), a través del cual fluido no puede pasar. Esto se logra haciendo que cuando una partícula colisione con dicho obstáculo la misma se refleja en la dirección en la que vino (con el vector dirección rotado 180 grados).

El siguiente paso es el calculo de colisiones en cada posición (y la propagación) de las partículas. Este proceso se hace en cada paso hasta completar la cantidad de pasos pedidos.

```
public void simulate(final int steps) {
    for (int t = 0; t < steps; t++) {
        solveCollisions();
        moveParticles();
        if (t % 4 == 0) {
            generateParticles();
        }
        if (t % velocityTimeStep == 0) {
            generateOutput();
        }
    }
    calculateReynoldNumber();
    calculateFlow();
}
```

Para el calculo de colisiones se suman los valores del arreglo de los distintos nodos. Esta suma indica la cantidad de partículas que colisionan en el nodo. De esta manera se distinguen los casos de colisiones entre 2 y 3 partículas. Una vez chequeados estos casos (y rotados según corresponda), se considera el caso de colisión con un obstáculo.

Finalmente se desplazan las partículas usando un mapeo de grilla bidimensional a grilla hexagonal ya que el mapeo varía si la coordenada  $y$  es par o impar.

Para la generación del output se subdivide el dominio en espacios más pequeños y se promedian las velocidades. De esta manera se reduce el ruido estadístico generado por las rotaciones pseudoaleatorias de las colisiones de 2 partículas y reduciendo el tiempo computacional/espacio del output (aunque sacrificando precisión a cambio). Ver figura 2.

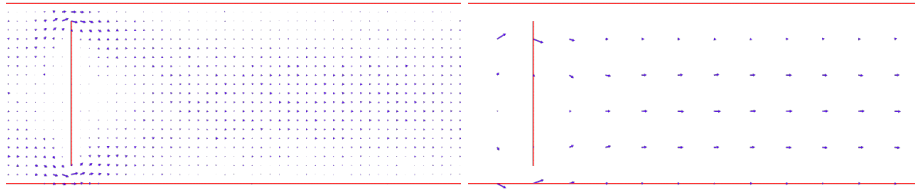


Figura 2: *Distintos tamaños de subdominio. Para tamaños más grande de subdominio se pierde precisión pero se reduce el ruido estadístico (a) Subdominio de  $16 \times 16 \text{ m}^2$ . (b) Subdominio de  $32 \times 32 \text{ m}^2$*

## 4. Resultados

A partir de la siguiente configuración se realizaron varias simulaciones:

- Tamaño dominio:  $960 \times 320 \text{ m}^2$
- Tamaño subdominio:  $16 \times 16 \text{ m}^2$
- $L$ : 32, 128, 256  $m$

- Frecuencia de generación del output: cada 100 pasos
- Cantidad de pasos de simulación: 10000 pasos

Las pruebas fueron ejecutadas en la siguiente PC:

- CPU: Intel i5-3337U @ 1.8GHz x2
- RAM: 4GB DDR3

En el cuadro 1 mostramos el resultados de correr cada prueba 10 veces, mostrando los valores promedio:

L (m)	Caudal ( $m^3/s^2$ )	Número de Reynold	Tiempo de ejecución(seg)
32	75600.72	2.22	227.02
128	57567.36	15.45	231.73
256	17438.00	19.78	249.38

Cuadro 1:  $Q$ ,  $Re$  y tiempos de ejecución en función de  $L$

A partir de esta tabla vemos que crece el número de Reynold a medida que se incrementa el tamaño de la barrera. Esto es coherente con la formula del número de Reynold:

$$Re = \frac{u_{prom} * L}{v(\rho)}$$

Donde  $u_{prom}$  es la magnitud de la velocidad promedio,  $L$  el tamaño de la barrera y  $v$  la viscosidad cinemática característica. Esto significa que crece la turbulencia del fluido. Sin embargo, por ser menor a 2100 el flujo se mantiene estacionario y se comporta como si estuviera formado por láminas delgadas, que interactúan sólo en función de los esfuerzos tangenciales existentes.

Por lo tanto para obtener niveles más grandes de turbulencia es necesario: incrementar la velocidad promedio, el tamaño de la barrera o disminuir la viscosidad que es una característica del fluido (en nuestro caso, es función de la densidad de la partículas). Esto se puede ver en la figura 3 de las configuraciones finales de los sistemas simulados en la que se nota una ligera diferencia en como se distribuyen las partículas con respecto a la barrera.

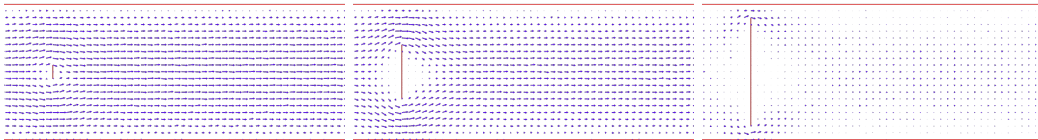


Figura 3: *Turbulencia alrededor del obstáculo para distinto  $L$ . (a)  $L = 32$  m (b)  $L = 128$  m (c)  $L = 256$  m.*

Por otro lado si miramos el caudal podemos ver que este disminuye a medida que incrementa  $L$ . Si analizamos la formula que describe el caudal:

$$Q = u_{prom} * A$$

Donde  $Q$  es el caudal,  $u_{prom}$  la velocidad promedio y  $A$  es el área de un corte perpendicular a la superficie del fluido (asumimos que la configuración simulada se correspondía a la de un cilindro, por lo que  $A$  es el área de un círculo determinado por la altura pasada como parámetro, el cual será su diámetro). Teniendo en cuenta que dicha área es constante para los distintos  $L$ , esta claro que la velocidad promedio disminuye.

Si graficamos la cantidad de colisiones contra la magnitud promedio de la velocidad (figura 4) notamos que si bien aumentan la cantidad de colisiones con el  $L$  (pues tienen menos espacio y ni bien salen del lado izquierdo chocan con la barrera), las velocidades disminuyen a medida que incrementamos el  $L$ . Esto se verifica viendo la figura 3 en la cual se nota que la magnitud de

los vectores velocidad disminuyen. Si bien no realizamos gráficos de la densidad, asumimos que la disminución de la velocidad se debe al incremento de la densidad de las partículas (lo cual es coherente con la disminución de la viscosidad e incremento del número de Reynold).

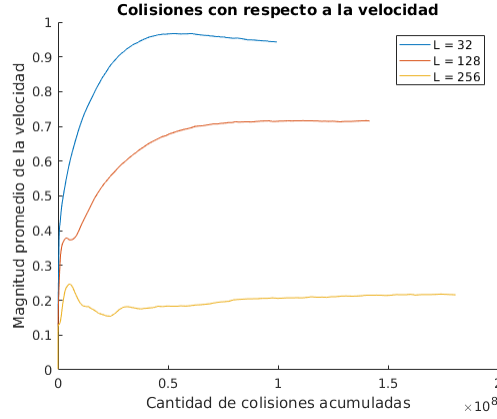


Figura 4: Velocidad en función de la cantidad de colisiones.

Probando un tamaño de dominio más grande, se observó un incremento significativo de la turbulencia (así como su tiempo de ejecución). Ver cuadro 2.

- Tamaño dominio:  $3072 \times 1024 \text{ m}^2$
- Tamaño subdominio:  $64 \times 64 \text{ m}^2$
- L:  $256 \text{ m}$
- Frecuencia de generación del output: cada 300 pasos
- Cantidad de pasos de simulación: 30000 pasos

Caudal ( $\text{m}^3/\text{s}^2$ )	Número de Reynold	Tiempo de ejecución(seg)
902566.93	24.09	14191.39

Cuadro 2: Resultados para un dominio de  $3072 \times 1024$

Viendo el valor  $R_e$  y la figura 5 se observa que ha aumentado (aunque no mucho) la turbulencia alrededor de barrera. A su vez, por ser el dominio más grande ha aumentando significativamente el caudal debido al incremento de área.

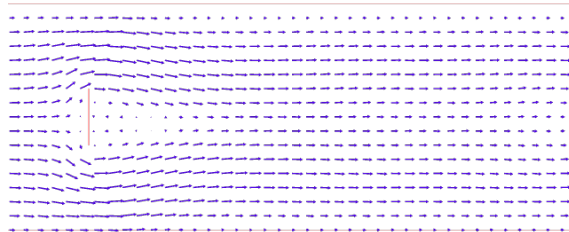


Figura 5: Configuración final para un dominio grande y 30000 pasos de simulación.

## 5. Conclusiones

Si bien el modelo de FPH-I es conveniente, sufre de un principal problema: la pérdida de precisión para dominios pequeños.

Si bien este problema es fácilmente solucionable tomando un tamaño de dominio más grande y más pasos de simulación, desafortunadamente esto requiere el uso una cantidad mayor de recursos (ya que habrá más colisiones).

De esto se desprende que de aumentar el tamaño del dominio requerirá más RAM y tiempo de ejecución. Para mejorar el algoritmo podrían hacerse mejoras como paralelizar el procesamiento de colisiones a través de multithreading (o incluso sobre GPU's).

A modo de conclusión, podemos decir que se implementó satisfactoriamente el algoritmo del autómatas celular FHP-I para la simulación de fluidos.

## Referencias

- [1] Dieter A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models - An introduction*, June 26, 2005.